

Some Interesting Things learned at CHEP[‡]

Robert K. Kutschke
June 9 /97

Outline:

- Treatment of Non-Gaussian Effects in track fitting.
- Comments on GEANT 4
- Comments on the Transition to C++.

[‡] while not really there.

Treatment of Non-Gaussian Measurement Errors

- All (almost all??) track fitters used in HEP production code assume gaussian errors on measurements, gaussian scatters at scattering surfaces, and gaussian straggling in energy loss.
- Of course this is not true.
- Invoke the the central limit theorem, burnt offerings etc to explain why we can get away with this.
- This gives rise both to biases in fitted results and to improperly determined errors — ie $\pm 1\sigma$ is not really the 67% confidence interval.
- Usual practice is to fudge the measurment errors to cover things — the net result is that we degrade the resolution on physics quantities, such as masses, lifetimes etc.
- Formally it is straightforward to allow for an arbitrary resolution function and an arbitrary scattering function.
- Practically it is very difficult: CPU requirements rapidly become prohibitive.
- Practical solutions have been suggested independently by Frühwirth (A169) and Brown (A341).
- Both papers are available from the conference web page:
<http://www.ifh.de/CHEP97/chep97.html>
- Both use Kalman filter formalism.

Conceptual Outline of a Kalman Filter

- 1) Initialize the state vector (ie track parameters and covariance matrix). Typically we do this at the position of the outermost hit on the track.
- 2) Add the information from the current hit to the track.
- 3) If this is the last hit, go to 5.
- 4) Transport the state vector from its present position in space to the next hit on the track. In the presence of material this causes loss of information. Go to 2.
- 5) Extrapolate the track to the beam line. In the presence of material this causes loss of information.

- Brown discusses non-gaussian measurement errors (step 2).
- Frühwirth discusses non-gaussian distribution of straggling in energy loss (steps 4 and 5).
- In the end, both methods are remarkably similar.

Step 2 Elaborated

Given:

- η = track parameters (eg. $\kappa, \phi_0, d_0, \tan \lambda, z_0$)
- V the covariance matrix of η .
- $\Delta\eta$ = the improvement to the track parameters from the information at this step.
- A measurement at this step, d_m and its error σ .

Find $\Delta\eta$ which maximizes the likelihood function,

$$\mathcal{L} = \mathcal{L}(\text{meas}) \mathcal{L}(\text{extrapolate}).$$

Traditionally we use,

$$\begin{aligned}\mathcal{L}(\text{meas}) &= \exp\left(-\frac{(d_m - d_f(\eta + \Delta\eta))^2}{\sigma^2}\right) \\ \mathcal{L}(\text{extrapolate}) &= \exp((- \Delta\eta)^T V^{-1}(\Delta\eta))\end{aligned}$$

In general,

$\mathcal{L}(\text{meas})$ is the pdf that $d_m - d(\eta + \Delta\eta) = 0$.

$\mathcal{L}(\text{extrapolate})$ is the pdf that $\Delta\eta = 0$.

We could define arbitrary pdf's and use MINUIT to minimize $-\log \mathcal{L}$ at each measurement!

Brown's Improvement to Step 2

- Consider a track with N hits.
- Generalize the resolution function to a sum of M gaussians, with different weights, $A_i, i = 1, M$.
- In principle there are $M \times N$ different fits which can be done and the optimal answer is some weighted average of all of these fits, where the weights are somehow related to the A_i .
- It is not practical to do all of these fits.
- Provided $A_{i+1} \ll A_i$, one can truncate this sum and still get a reliable answer.
- Once you have done one full fit and determined the residuals, you have some idea of which A_i are important for each hit.
- Brown developed a truncation algorithm which typically keeps a few terms.
- The result is that for a factor of 2 cost of CPU time, the track parameter errors can be reduced by 10% of themselves. After this, improvement comes much more slowly.

Früwirth's Improvement to Step 4

- Each propagation step will pass through material.
- The amount of energy loss in each step is described by a sum of gaussians, say that 2 gaussians are enough.
- One propagation step gives 2 tracks.
- Next propagation gives 4 tracks.
- Next propagation gives 8 tracks.
- At this point he retains the 6 most probable tracks. The number 6 is was determined empirically.
- The next step gives 12 tracks.
- At this point he retains the 6 most probable tracks.
- The net result is that there are improvements of order 10% on track parameter errors for a cost of a factor of 2 in CPU time.
- Unlike Brown, this is a single pass affair.
- He has shown that the error is a formally well defined quantity. Brown has not done this analysis.

A Few Comments on GEANT 4

- I had several conversations with Simone Gianni about the timetable for G4.
- The alpha release is now out but only to developers, where “developers” is very tightly defined. For example, there is no documentation at all about how to use the tracking code — so only groups with an active participate in the tracking code development will have access to the alpha release of the tracking code.
- PAT is collaborating on the G4 graphics — therefore FNAL does have access to the alpha release, but only for graphics purposes. We do not have any in house G4 tracking expertise and so to not have “user” access to the alpha release.
- The timetable for the beta release, which will have documentation, and which will be widely available, is still scheduled for May 1998.

Sometimes I Redesign the Class Library 3 times before Breakfast

- Some proponents of OOAD claim that one can design classes at the outset sufficiently well that those writing the methods do not need to change the class structure.
- Other proponents claim that the processes of class design and method development is intrinsically iterative.
- Reports from BaBar tracking people are that they find it to be intrinsically iterative:
 - As they understand more and more about pattern recognition and track fitting, they need to change the relationship among various objects.
 - It is not possible to understand these requirements until you actually start to develop the methods.
 - This has consequences for external data representations, persistence etc.
 - It occasionally has consequences for other parts of the code that need access to tracking information.
- This is not a serious problem since a sufficiently small number of people are working on these parts of the code and all are capable of using the tools which make updates easy.

My own feeling is that, for most large projects, we all will find the problem to be intrinsically iterative and that allowance for these iterations must be built into designs.